PATENT
Atty. Dkt. No. YOR920030510US1

IN THE SPECIFICATION

I. Please replace paragraph [0002] as follows:

Since the advent of the first electronic computers in the 1940s, computers have expanded from performing complex numeric calculations, to processing increasingly complex symbolic programming languages (~~the~~ FORTRAN, COBOL, ~~Pascal~~ PASCAL, C, C++, ~~the~~ and JAVA programming ~~language~~ languages, and the like), to data representing text, documents, images and speech. During this same period, programming approaches have evolved from assembly language (binary machine code) to structured programming techniques, and finally to current object-oriented approaches as represented by the SMALLTALK, C++ and JAVA programming languages. These programming languages require programmers to specify the sequences of operations necessary to perform the data processing functions of the application, and are referred to as procedural languages.

II. Please replace paragraph [0043] as follows:

A ruleset author can create rulesets and templates using the ARL ~~Swing RuleSet~~ SWING RULESET editor or the WEBSPHERE ~~Studio Application Developer~~ STUDIO APPLICATION DEVELOPER ARL ~~Editor~~ EDITOR plugin, commercially available from IBM corporation of Armonk, New York. There are no external artifacts or special-purpose Graphical User Interfaces required to use the templates in the present invention. The ARL compiler parses and generates an AbleRuleSet bean with the template information contained in it. Afterward, an AbleRuleSet bean with templatized components can be customized with PC client or web-based user interfaces via the AbleRuleSet bean template application programming interfaces (APIs).

III. Please replace paragraph [0048] as follows:

In one embodiment, a text ruleset 210 comprises a set of rule language specifications, which can be parsed by the text parser 221 and compiled into a collection of objects, for example, variables 226, literals 227 and expressions 228. Using the rule APIs 225, these objects can be converted back into a text ruleset. An

XML ruleset 211 comprises a set of rule language specifications, which can be parsed by the XML parser 222 and compiled into a collection of objects. Using the rule APIs 225, these objects can be converted back into an equivalent XML ruleset. An application program 212 can construct a ruleset and a collection of objects using the APIs 225 provided with the framework. A valid ruleset, whether parsed using the text parser 221, the XML parser 222 or the rule APIs 225 results in the creation of an identical collection of objects. These methods provide equivalent capabilities of defining and constructing framework objects 226-233 226-231, 272, and 274.